1 Números de punto flotante

Un número de de punto flotante de base b, exceso q con p dígitos, se representa mediante el par ordenado de valores (e, f), denotando $(e, f) = f \times b^{e-q}$.

Aquí, e es un entero (con un rango preestablecido) y f es una fracción con signo.

En el caso de números binarios, $b = 2$.

Cuando se trabaja con números binarios, generalmente se utilizan números normalizados. Un número de punto flotante(e, f) está normalizado si el dígito más significativo de la representación es distinto de cero, de manera tal que $1/b \leq |f| < 1$ o $f = 0$ y e el menor valor posible. La ventaja de los números normalizados es que permiten eliminar posibles ambigüedades (varias formas de representar el mismo número). La principal desventaja es que las magnitudes pequeñas (e.g. 0.00000001) no pueden normalizarse sin producir exponentes negativos.

1.1 Errores

Es común, al tratar con números de punto flotante, medir el error de un número utilizando la diferencia relativa con respecto al número verdadero

$$Error\ relativo = \left| \frac{verdadero - calculado}{verdadero} \right|$$

Esta medida de error falla cuando el valor verdadero es cero, o cercano a cero. Por ejemplo, al calcular $\sin(\pi)$ tendríamos (suponiendo tres dígitos de punto flotante).

$$Error\ relativo = \left| \frac{\sin \pi - \sin\left(.314 \times 10^1\right)}{\sin \pi} \right|$$

y, puesto que $\sin(.314 \times 10^1)$ no es exactamente cero, el error relativo sería infinito. Por esta razón al calcular el error relativo es mejor utilizar como denominador $\max\left(|x|, |f(x)|\right)$ en lugar de $f(x)$.

1.2 Operaciones con números de punto flotante

El producto de dos números de p dígitos utilizando aritmética convencional es un número de $2p$ o $2p - 1$ dígitos, pero en el sistema de numeración de punto flotante con p dígitos sólo pueden utilizarse p dígitos. ¿Qué número se debe utilizar para indicar el resultado?. El sentido común sugiere utilizar el número de p dígitos más cercano al producto. En el caso ambiguo de que existen dos números de p dígitos igualmente cercanos al resultado, usualmente se elige el mayor número. Esto introducce un sesgo, pero en general el efecto de este sesgo es insignificante en la práctica.[1]

La división es un poco más complicada, pero el efecto del redondeo es el mismo. Se selecciona el número de p dígitos más cercano al cociente matemáticamente correcto con un ligero sesgo.

La suma y la resta requieren en primer lugar la comparación de los exponenetes, desplazando una parte fraccionaria con respecto a la otra antes de ejecutar la suma. Nótese que la resta puede producir muchos ceros (o sólo ceros) en el resultado.

[1] Esto no significa que el error de redondeo es insignificante. Lo que se considera insignificante en la práctica es la diferencia entre redondear siempre hacia arriba en lugar de hacerlo hacia arriba la mitad de las veces y hacia abajo la otra mitad.

1.3 Consideraciones importantes sobre números de punto flotante

1.3.1 Cálculo de sumas

Utilizar el siguiente algoritmo (Fórmula de suma de Kahan):

```
S=x[1]
c=0
para j=2 a N {
    y=x[j] - c;
    t=S+y;
    c=(t-S)-y;
    s=t;
}
```

1.3.2 Cálculo de medias y desviaciones estándar

```
M=x[1]
S=0
para i=2 a N {
    M_prev=M;
    M=M_prev+(x[i]-M_prev)/k
    S_prev=S;
    S=S_prev+(x[i]-M_prev)*(x[i]-M)
}
```

$M_1 = x_1, \;\; M_k = M_{k-1} \oplus (x_k \ominus M_{k-1}) \oslash k$

$S_1 = 0, \;\; S_k = S_{k-1} \oplus (x_k \ominus M_{k-1}) \otimes (x_k \ominus M_k)$

1.3.3 Comparaciones

En general, nunca se debe comparar si dos números de punto flotante son exactamente iguales (incluso, si desde el punto de vista teórica deberían serlo) debido a que esto es extramadamente improbable.

Por ejemplo, si se usa una relación de recurrencia $x_{n+1} = f(x_n)$ en la cual teóricamente x_n tiende a un límite a medida que $n \to \infty$, usualmente es un error esperar a que $x_{n+1} = x_n$ para cierto n, puesto que la secuencia x_n podría ser periódica debido al efecto del redondeo. El procedimiento apropiado es esperar hasta que $|x_{n+1} - x_n| < \delta$, para un valor valor conveniente δ; pero como no necesariamente se conoce el orden de magnitud de x_n es mejor aún esperar hasta que se cumpla

$$|x_{n+1} - x_n| \leq \epsilon \, |x_n|$$

ϵ es un número mucho más fácil de seleccionar.

El hecho de que el concepto de igualdad estricta es de poca importancia cuando se trabaja con números de punto flotante implica que que se debe definir una nueva operación, *comparación de punto flotante*, la cual se utiliza para evaluar los valores relativos de dos cantidades de punto flotante. Pueden utilizarse las siguientes definiciones para evaluar números de punto flotante u y v.

- $u \prec v \quad (\epsilon)$ si y solo si $v - u > \epsilon \max(u, v)$
- $u \sim v \quad (\epsilon)$ si y solo si $|v - u| \leq \epsilon \max(u, v)$
- $u \succ v \quad (\epsilon)$ si y solo si $u - v > \epsilon \max(u, v)$

- $u \approx v \quad (\epsilon)$ si y solo si $|v - u| > \epsilon \min(u, v)$

Nótese que para cualquier para de números de punto flotante u y v se cumple exactamente una de las condiciones $u \prec v$ (u es definitivamente menor que v), $u \sim v$(u es aproximadamente igual a v) y $u \succ v$(u es mayor que v). La relación $u \approx v$ es algo más fuerte que $u \sim v$ y podría leerse como "u es esencialmente igual a v". Todas estas relaciones están en función de un real positivo ϵ que mide el grado de aproximación considerado.

Nótese también que la definición de igualdad aproximada no es una relación de equivalencia.

1.4 Números binarios de punto flotante formato IEEE 754

El estándar IEEE 754 define un formato para números binarios de punto flotante que considera:

- Uso de números normalizados
- Uso de números no normalizados en ciertos casos (magnitudes pequeñas)
- Casos excepcionales (básicamente Infinitos y valores indeterminados como 0/0).

El formato sirve para registros binarios de cualquier tamaño, aunque especifica de manera precisa los casos de 32 y 64 bits.

El formato divide una palabra de n bits en tres partes, siguiendo el orden dado a continuación:

- 1 bit para el signo. Se denotará con s el valor del signo (0 o 1)
- p bits para el exponente. Se denotará con e el valor entero del exponente.
- q bits para la parte fraccionaria. Se denotará con f la secuencia de bits de la parte fraccionaria.

La siguiente table muestra los posibles valores de un número de punto flotante en formato IEE 754.

Patrón de bits	Valor	Comentarios
$0 < e < 2^p - 1$	$(-1)^s \times 1.f \times 2^{e-exceso}$	Números normalizados. El exceso es $2^{p-1} - 1$
$e = 0;\ f \neq 0$	$(-1)^s \times 0.f \times 2^{e-exceso}$	Números subnormalizados. El exceso es $2^{p-1} - 2$
$e = 0;\ f = 0$	$(-1)^s \times 0.0$	Cero con signo
$s = 0;\ e = 2^p - 1;\ f = 0$	+INF	Infinito positivo. Todos los bits del exponente en 1 y el resto en 0.
$s = 1;\ e = 2^p - 1;\ f = 0$	-INF	Infinito negativo. Todos los bits del exponente en 1, signo en 1 y parte fraccionaria en 0.
$e = 2^p - 1;\ f \neq 0$	NaN	No número. Valor indeterminado. Se produce al hacer operaciones como 0/0 o INF-INF. Todos los bits del exponente en 1 y al menos un bit de la parte fraccionaria en 1.

En el caso particular de IEEE 754 de precisión simple se utilizan palabras de 32 bits con 8 bits para el exponente y 23 bits para la parte fraccionaria. En lenguajes de programción, esto corresponde a float en Java, REAL*4 en Fortran y float en la mayoría de los compiladores de C y C++.

Para doble precisión se utilizan palabras de 64 bits con 11 bits para el exponente y 52 bits para la parte fraccionaria.